Learning to Extract Gene-Protein Names from Weakly-Labeled Text

Richard C. Wang, Anthony Tomasic, Robert E. Frederking,
Isaac Simmons, William W. Cohen

CMU-LTI-08-004

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2008		**00-00-2008 to 00-00-2008**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Learning to Extract Gene-Protein Names from Weakly-Labeled Text	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Carnegie Mellon University,School of Computer Science,5000 Forbes Ave,Pittsburgh,PA,15213**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Training a named entity recognizer (NER) has always been a difficult task due to the effort required to generate a significant amount of annotated training data. In this paper, we reduce or eliminate the effort required to create training data by automatically converting other sources of data into annotated training data. The performance of this approach is tested on a geneprotein name extractor by using the mouse and fly data obtained from the BioCreAtIvE challenge. Results show that our methods are effective and that our trained NER system outperforms all of our baseline results.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**11**	

Learning to Extract Gene-Protein Names from Weakly-Labeled Text

Richard C. Wang[1], **Anthony Tomasic**[2], **Robert E. Frederking**[1]
Isaac Simmons[2], **William W. Cohen**[3]

[1] Language Technologies Institute
[2] Institute for Software Research International
[3] Machine Learning Department

Carnegie Mellon University, 5000 Forbes Ave.
Pittsburgh, PA 15213, U.S.A.

Abstract

Training a named entity recognizer (NER) has always been a difficult task due to the effort required to generate a significant amount of annotated training data. In this paper, we reduce or eliminate the effort required to create training data by automatically converting other sources of data into annotated training data. The performance of this approach is tested on a gene-protein name extractor by using the mouse and fly data obtained from the BioCreAtIvE challenge. Results show that our methods are effective and that our trained NER system outperforms all of our baseline results.

1 Introduction

Many prior research papers on biological text-mining have developed machine-learned *named entity recognition* (NER) systems to identify substrings in biomedical publications that correspond to gene and protein names, usually without distinguishing between them [4, 9, 11, 16]. These NER systems are often trained on large amounts of manually annotated training examples, consisting of documents with the position of every named entity marked. This training data is difficult to produce.

Training data for gene-protein entities is especially difficult to produce because labeling documents requires expertise in biology. Although a number of corpora have been annotated, the documents in these corpora are drawn from specific sub-areas of biology. Here we consider two such corpora: the YAPEX[1] [10] training corpus, which consists of Medline abstracts selected as likely to contain information about protein-protein interactions; and the GENIA[2] [7] corpus, which contains abstracts likely to contain information about cell signaling in human blood cells. As we will show, extractors trained on these corpora appear to be distribution-specific (i.e. they do not transfer well to other sub-areas of biology, or different genres of text within the same sub-area).

The distribution-specificity of learned NER systems makes it difficult to use them in certain types of text-mining systems. As an example, consider the SLIF system [23], which mines full-text biomedical publications for information about subcellular localization of proteins. More specifically, SLIF finds figures containing images of a certain sort (fluorescence microscope images depicting protein localization), and then collects, analyzes and indexes these figures by the proteins depicted. For this application it is necessary to apply NER methods to figure captions; however, the majority of NER training sets are annotated abstracts.

Motivated by such problems, this paper explores several approaches for training a gene-protein NER system with data sources that are often easier to obtain. The first source is NER annotations for a related, but slightly different corpus: this reflects the common practice of applying a learned NER system to documents that are drawn

[1] Available from http://www.sics.se/humle/projects/prothalt
[2] Available from http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA

Dataset	Mouse			Fly		
Data	**Eval.**	**Weak-train**	**Curated**	**Eval.**	**Weak-train**	**Curated**
# of Abstracts	50	200	1000	51	57	1000
Abstract IDs	100-149	1-99, 150-250	4000-4999	[1-298]	[308-494]	4000-4999

Table 1. Distribution of abstracts among various data for each of mouse and fly dataset. Numbers embraced by brackets indicate a subset of these numbers.

from a slightly different distribution. The second source is a *synonym list* – a list of gene identifiers together with synonyms for each identifier. The third source is *weak labels*, which associate a document with identifiers for each gene-protein entity that appears in the document. Weak labels for text can often be automatically obtained by analyzing databases of information extracted from text. Specifically, weak labels are often obtainable for biomedical documents by analysis of manually-curated biological knowledge bases such as FlyBase [1] and MGI [2].

One prior experimental study that exploits synonym lists and weak labels is BioCreAtIvE task 1B [14, 15], which collected common test-bed problems and a common evaluation framework for determining the database identifier of every gene mentioned in biomedical abstracts – a task closely related to NER, but distinct. Three separate test-bed problems were developed, one for each of three model organisms: yeast, fly, and mouse.

In this paper, we utilize only the mouse and fly datasets, which were the two hardest for the BioCreAtIvE participants, for training a gene-protein NER system. Performance on NER is evaluated by testing on a small subset of the BioCreAtivE test set that was manually annotated. We compare weakly-learned NER systems with results for four baseline systems. The first baseline is a dictionary-based extractor, which soft-matches words from a synonym list to a corpus. The second, third, and fourth baselines are machine-learned NER systems trained on the GENIA dataset, the YAPEX dataset, and small corpora of conventionally-labeled documents from the BioCreAtive datasets.

Experimentally, we show that no baseline system performs well on the evaluation data – the best baseline F_1 measures reach only 57% on the mouse data, and 41% on the fly data. We then present results for several alternative approaches that use weak labels, and demonstrate that much better performance can be obtained with weakly-trained NER systems.

Our approach for weak-label learning consists of four steps. First, we look up, for each abstract, its associated gene identifiers and we label all possible locations of synonyms associated with these identifiers in that same abstract. Second, we train extractors on these weakly labeled abstracts, using word features such as string similarity to synonyms [24]. We also investigated a pre-processing step, of removing from the training set sentences not containing any weak labels; and a post-processing step that exploits inter-document repetition of names [20] by soft-matching every instance of an extracted name against the document in which it occurs, and classifying every such soft-match as a protein name. To further evaluate our weak-label learning approach, we present also results for NER systems tuned for either precision or recall [21]. Our results show that the quality of a NER system can be improved through the use of readily available weakly-labeled data.

We use datasets from BioCreAtIvE task 1B, specifically the mouse and fly datasets, which were drawn from MGI and FlyBase respectively. For each dataset, we constructed three corpora for our experiments: *evaluation*, *weak-train*, and *curated*. The *evaluation* and *weak-train* data are subsets of the BioCreAtIvE "devtest" set, and *curated* data is a subset of the "training" set. Table 1 summarizes, for both datasets, their size, and also lists the specific abstracts (by BioCreAtIvE ID) that were used to form the dataset. In *curated*, each abstract is associated with gene identifiers of all genes that are mentioned in the full text of the abstract. However, in *weak-train*, each abstract is only associated with identifiers of some genes mentioned in the abstract. Hence, *curated* is noisier than *weak-train*. We also utilize the synonym lists provided by the mouse and fly datasets, which contain associations between synonyms and unique gene identifiers. The list for the mouse dataset consists of 183,142

synonyms for 52,594 identifiers, and for the fly dataset, 135,471 synonyms for 35,970 identifiers. To evaluate our NER systems, the abstracts in the evaluation data were manually annotated with gene-protein entity names.

2 Baseline Methods

2.1 Global Edit Distance

In order to train a gene-protein NER system using a synonym list, we devise a feature that indicates how similar each word in the abstracts is to the most similar word in the entire (global) synonym list. The similarity measure incorporates Levenshtein Distance [19], and thus we call this the *global edit distance* (GED) feature. Elsewhere it has been shown that features of this sort can substantially improve NER performance [24].

More specifically, GED case-insensitively calculates a similarity score between two strings, *s* and *s'*, as:

$$SimScore(s, s') = 1 - \frac{LD(s, s')}{\max(length(s), length(s'))} \quad (1)$$

where *LD*(*s*, *s'*) is the Levenshtein Distance between string *s* and *s'*, and *length*(*s*) is the number of characters in *s*. We determine and assign similarity scores to each word in the abstracts by traversing through each synonym in a given list. For each synonym *s*, we determine number of words *n* contained in *s*, and create sliding windows of size ranging from $\lceil 0.5n \rceil$ to $\lfloor 1.5n \rfloor$ on the abstract. For each string *s'* contained within each sliding window, we assign *SimScore*(*s*, *s'*) to each word *w* in *s'* unless one of the following conditions is met: a) *w* has higher similarity to some other *s"* in the synonym list, b) *s* or *s'* has only one character, c) *s* or *s'* case-insensitively matches any word in a list of common stop-words (see Appendix A), or d) the first and last characters of *s* are not identical to those of *s'*.

2.2 Soft Matching

Biological scientists often use novel variations of existing gene names in their papers; thus, in order to match these names from abstracts to the synonym list, we incorporate an approximate string matching technique called *soft matching*, which identifies strings that are similar but not necessarily identical. This method has been proven to be useful [13]; however, our method is on the character-level instead of word-level. Our soft matching is performed as follows: First, we assign similarity scores to words in given abstracts using a given synonym list, as described in 2.1. We then label all the longest consecutive sequences of words that have similarity scores above a given similarity threshold as a gene-protein entity name.

2.3 NER on YAPEX & GENIA

We use an off-the-shelf machine learning system for NER called Minorthird [6] for training our gene-protein NER system on the YAPEX and GENIA corpora. We used Minorthird's default feature set, which contains basic features such as word identity and capitalization patterns. In addition, we used Minorthird's implementation of VP-HMM – a voted-perceptron based training scheme for HMMs due to Collins [8]. VP-HMM is generally competitive with conditional random field (CRF) learning methods, but converges more quickly. More specifically, as we configured this learner, NER is reduced to the problem of classifying each token as the *beginning* or *continuation* of a multi-token gene-protein name; or as *outside* of any gene-protein name. We configured the extractor to make 20 passes (epochs) over the training data using a window size of three words.

The YAPEX dataset consists of a training corpus of 99 Medline abstracts and a testing corpus of 101 Medline abstracts. These documents deal primarily with protein-protein interactions, and are annotated for gene-protein entities. We trained a VP-HMM extractor on the training corpus of YAPEX using Minorthird's default features. The GENIA dataset consists of a training corpus of 500 Medline abstracts and a testing corpus of 300 Medline abstracts, mostly concerning cell signaling for human cells. We trained a VP-HMM extractor on the training corpus of GENIA using default features, plus protein-specific features described elsewhere [18].

2.4 Single Document Repetition

When a substring is identified as a named entity in a document, it is highly possible that all other occurrences of that substring in the same document are also named entities. Repetition of names in text

	Mouse			Fly		
	Entity F_1	**Δ Baseline**	**Δ Complete**	**Entity F_1**	**Δ Baseline**	**Δ Complete**
Best Baseline	57.64	-	-19.28%	45.75	-	-26.09%
Complete System	71.41	23.89%	-	61.90	35.30%	-
Complete - Weak-train	71.12	23.39%	-0.41%	63.02	37.75%	1.81%
Complete - Filter	70.29	21.95%	-1.57%	63.54	38.89%	2.65%
Complete - SDR	67.45	17.02%	-5.55%	65.39	42.93%	5.64%
Complete - GED	66.67	15.67%	-6.64%	57.14	24.90%	-7.69%
Best System 1: (Complete)	71.41	23.89%	-	61.90	35.30%	-
Best System 2: (Complete - Weak-train - Filter - SDR)	60.39	4.77%	-15.43%	66.41	45.16%	7.29%

Table 2. Summary of the performance of our best baseline system and various configurations of our complete system for each of mouse and fly dataset. Configurations are derived by subtracting various components from the complete system. Detailed results are presented in Table 3 for the mouse and Table 4 for the fly.

has proven useful on many occasions [3, 17, 20, 25]. We incorporate a post-processing step that exploits repetition of entity names within a single document using the gene-protein names extracted by our trained NER systems. More specifically, for each abstract, it collects all the extracted names from that abstract, and soft-matches these names against the words in the same abstract, using a constant threshold of 0.5 throughout our experiments; we refer to this as single document repetition (SDR) labeling.

3 Approach

3.1 Grounding Weak Labels

In the BioCreAtIvE challenge, one unique characteristic of the datasets is that there are synonym lists and weak labels. Therefore, for each abstract, we can approximately locate gene names by soft-matching synonyms of identifiers associated with that abstract against the words in the same abstracts. For this process, we used the fixed similarity threshold of 0.5 for both the mouse and the fly datasets; we will refer to this process as *grounding* the weak labels. The result of grounding is a set of documents that are noisily annotated; a preliminary evaluation of our grounding method on the evaluation data shows that grounding gives an entity-level precision of 81%, recall of 65%, and F_1 of 72% for the mouse dataset, and precision of 73%, recall of 70%, and F_1 of 71% for the fly dataset.

3.2 Sentence Filtering

Often genes that are mentioned but not associated with new results are not curated. Sections of a document that discuss these genes will become false negatives in our training set – i.e., they contain substrings that should be annotated as protein names, but are not. One method for eliminating (some of) these false negatives is to filter out portions of the document that are likely to contain false negatives. We thus incorporate a pre-processing step of filtering training examples: specifically, we split abstracts into sentences (using a regular expression), and then remove sentences in the training data that do not contain any grounded gene-protein synonyms. We call this the *sentence filtering* process. Recently, the same sentence filtering technique was independently described by Vlachos and Gasperin [26].

Sentence-filtering will also remove many true negative examples; hence, one might expect that sentence-filtering would lead to an over-general extractor, and hence increase recall at the expense of precision. Section 5 discusses methods to compensate for this bias.

4 Experiments

4.1 Settings

We trained a VP-HMM extractor on each of the following three datasets: *weak-train*, *curated*, and a combined set, *merged*, which is the union of *cu-*

			-SDR			+SDR		
			Entity Prec.	**Entity Recall**	**Entity F_1**	**Entity Prec.**	**Entity Recall**	**Entity F_1**
		YAPEX	68.36	27.56	39.29	69.28	48.29	56.91
		GENIA	66.46	24.37	35.67	67.45	39.18	49.57
		Dictionary	50.34	67.43	57.64	47.56	66.51	55.46
	C.V. Eval.	**-GED**	54.81	29.84	38.64	49.28	38.95	43.51
		+GED	59.05	53.53	56.15	54.75	60.36	57.42
	Weak-train	**-GED**	82.39	26.65	40.28	78.76	34.62	48.10
		+GED	78.47	48.97	**60.31**	75.58	59.23	**66.41**
Curated	**-GED**	**-Filter**	90.82	20.27	33.15	90.96	34.40	49.92
		+Filter	74.67	38.27	50.60	71.97	60.82	**65.93**
	+GED	**-Filter**	87.83	46.01	**60.39**	83.59	61.50	**70.87**
		+Filter	80.91	56.95	**66.84**	76.10	66.74	**71.12**
Merged	**-GED**	**-Filter**	90.35	23.46	37.25	78.63	41.91	54.68
		+Filter*	78.30	41.91	54.60	73.10	61.28	**66.67**
	+GED	**-Filter**	87.40	50.57	**64.07**	84.13	60.36	**70.29**
		+Filter*	79.57	58.54	**67.45**	75.90	67.43	**71.41**

Table 3. Performance of the four baselines (YAPEX, GENIA, Dictionary, and C.V. Eval.) and our NER systems (Weak-train, Curated, Merged) at entity-level tested on the **mouse** evaluation data. Bold F_1 scores represent scores that are higher than any corresponding baseline. Extractors denoted by * will be tuned in section 5.

rated and *weak-train*. Each of these datasets is weakly-labeled with grounded gene-protein synonyms, using the approach described in 3.1. Each trained extractor is evaluated with various combinations of sentence filtering, SDR labeling, and GED features. These extractors are evaluated on the evaluation data at the entity-level (i.e., no partial credit is given for nearly-correct entity boundaries).

We compare our NER system's performance to four baselines: a) an extractor trained on YAPEX, b) another trained on GENIA, c) 10-fold cross validation on the evaluation data, and d) a global dictionary soft-matcher which soft-matches every synonym from an entire synonym list to the evaluation data (exact-matching was found to perform worse). The similarity thresholds[3] of the soft-matcher were pre-determined to optimize F_1 measure on the evaluation data, so they are optimistic assessments of the performance of this sort of technique. In addition to the four baseline performances, we present our NER systems performance at the entity-level in Table 3 and 4.

[3] Specifically, they are 0.85 for mouse and 0.95 for fly dataset

4.2 Results

None of the baseline methods is competitive with the complete system (including GED features, SDR, and sentence filtering) trained on the largest weakly-labeled dataset (merged). Table 2 shows a summary of our experimental results. For mouse, the complete system obtains an F_1 of 71.4% and the best baseline (soft-match to the dictionary) obtains an F_1 of 57.6%; for fly, the complete system obtains an F_1 of 61.9%, and the best baseline (a YAPEX-trained system) obtains F_1 of only 45.8%. Table 2 also shows the relative improvement in F_1 between the complete systems and the best baseline – the improvement is nearly 24% for mouse, and more than 35% for fly.

Table 2 also shows the results for training on only the *curated* data (in the row labeled "Complete - Weak-train"); for training without sentence-filtering (row "Complete - Filter"); for training without SDR; and for training without the GED features. Each of these ablations performs worse on the *mouse* data, although the effects are small for "Complete - Filter" and "Complete - Weak-train". For fly, the trends are less clear: removing the GED features clearly leads to lower performance, but removing SDR results in noticeably higher performance, and removing sentence-

			-SDR			+SDR		
			Entity Prec.	**Entity Recall**	**Entity F_1**	**Entity Prec.**	**Entity Recall**	**Entity F_1**
		YAPEX	66.00	23.32	34.46	68.79	34.28	45.75
		GENIA	44.16	12.01	18.89	59.06	26.50	36.59
		Dictionary	28.92	70.32	40.99	27.75	70.32	39.80
	C.V. Eval.	**-GED**	39.13	9.54	15.34	46.38	22.61	30.40
		+GED	37.59	36.40	36.98	35.68	46.64	40.43
	Weak-train	**-GED**	37.50	3.18	5.86	38.46	7.07	11.94
		+GED	51.89	38.87	**44.44**	56.79	57.60	**57.19**
Curated	**-GED**	**-Filter**	78.43	28.27	**41.56**	75.71	47.35	**58.26**
		+Filter	63.64	44.52	**52.39**	55.25	57.60	**56.40**
	+GED	**-Filter**	73.19	60.78	**66.41**	64.31	64.31	**64.31**
		+Filter	65.73	66.43	**66.08**	57.82	69.26	**63.02**
Merged	**-GED**	**-Filter**	78.26	31.80	**45.23**	74.30	47.00	**57.58**
		+Filter*	64.47	44.88	**52.92**	55.70	58.66	**57.14**
	+GED	**-Filter**	70.76	59.01	**64.35**	62.46	64.66	**63.54**
		+Filter*	64.38	66.43	**65.39**	56.20	68.90	**61.90**

Table 4. Performance of the four baselines (YAPEX, GENIA, Dictionary, and C.V. Eval.) and our NER systems (Weak-train, Curated, Merged) at entity-level tested on the **fly** evaluation data. Bold F_1 scores represent scores that are higher than any corresponding baseline. Extractors denoted by * will be tuned in section 5.

filtering or the (57 document) *weak-train* dataset also leads to slight improvements in performance.

The last two rows of Table 2 report performance of the system that uses the best combination of techniques, as suggested by these ablation studies. For mouse, this is the complete system; for fly, it is the system trained on the *curated* data only, with GED features, but without SDR and sentence filtering. This system achieves a 45% improvement over the best baseline.

Tables 3 and 4 also show the result of every combined system. In the mouse dataset, the weakly-trained NER systems outperform the best baseline whenever they are trained with GED, or whenever it included sentence-filtering and SDR. For the fly dataset, our NER systems almost always outperform all baselines. For the mouse dataset, filtering, SDR, and GED always improve F_1, and the maximum F_1 measure of 71.4% is obtained when all three methods are combined. For the fly dataset, only GED is always effective, and SDR is effective only when not combined with GED. We conjecture that when precision is high and recall is low, SDR is more likely to label false negatives than true negatives as gene-protein names.

The maximum F_1 score on the fly dataset was obtained on the unfiltered curated data; however, the performance of the nearly-complete system (with GED and filtering) trained on the largest (merged) dataset is similar (65.4%). We conjecture that for future weak-training problems competitive performance can be obtained by either the complete system, or the complete system without SDR.

5 Extractor Tuning

5.1 Method

The sentence-filtering method described above increases recall at the expense of precision, which may not be appropriate for all text-mining applications. In general, one would like for it to be possible to adjust the recall-precision tradeoff of an NER system to suite the user's need; for instance, curators of biological databases might prefer a high-recall gene-protein name extractor to assist them in identifying most gene-protein candidate names. To create such an extractor we tune or *tweak* [21] the threshold term of some of our trained extractors (those marked with * in Table 3 and 4) on the word-level recall of the tuning data *weak-train* (which is less noisy than *curated*). We pick the threshold term to optimize a user-chosen β value in the complete F-measure formula:

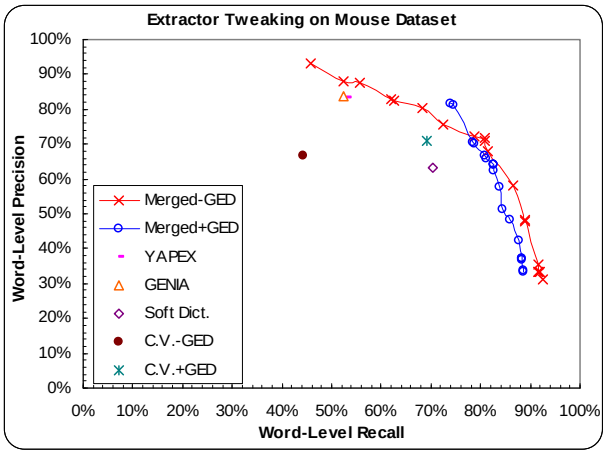

Figure 1. Tweaking extractors trained on the **mouse** dataset for β values from 0.1 to 10 on the *word-level* recall of *weak-train* data. The four baselines are also shown. *Merged* was filtered, and all extractions were SDR labeled.

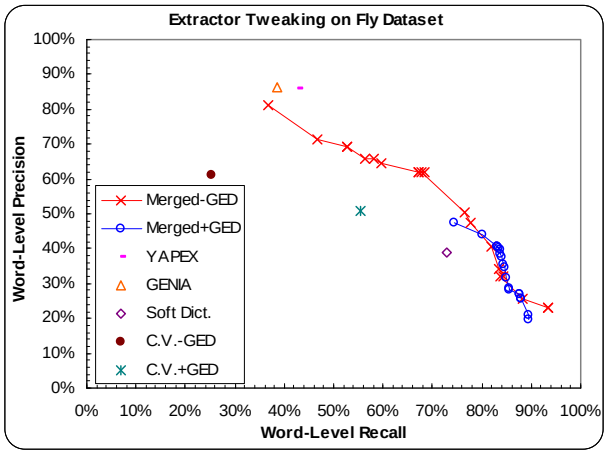

Figure 2. Tweaking extractors trained on the **fly** dataset for β values from 0.1 to 10 on the *word-level* recall of *weak-train* data. The four baselines are also shown. *Merged* was filtered, and all extractions were SDR labeled.

$$F_\beta(P,R) = \frac{(\beta^2+1)PR}{\beta^2 P + R} \quad (2)$$

Here P is word-level precision and R is word-level recall. A β value of greater than 1 assigns higher importance to recall; for instance, F_2 weights recall twice as much as precision. These tweaked extractors are then evaluated on the evaluation data.

Word-level precision measures the fraction of words (tokens) that are part of a predicted entity name, relative to the number of words that are part of an actual entity name. Use of word-level precision and recall rather than entity-level precision and recall gives some credit to nearly-correct entity boundaries – for instance, an extractor that extends slightly past an entity boundary will receive credit for word recall, but be penalized for word precision.

5.2 Results

In Figure 1 (mouse) and 2 (fly), each shows two precision-recall curves at the word-level; one is a curve of tweaked extractors trained without GED features and the other with GED features. Each data point on a line represents an extractor tweaked for a different β value (0.1, 0.2, …, 0.9, 1, 2, …, 10) trained on filtered examples and has extractions SDR labeled.

As comparisons, we also show the four baselines: a YAPEX-trained NER system, a GENIA-trained NER system, soft matching using dictionary, and 10-fold cross validation (with and without GED features). As expected, the higher the β value, the higher the word-level recall of the resulting tweaked extractor. Interestingly, while including the GED features always improves F_1, it also appears to limit the degree to which precision can be traded off for recall. We were able to generate a high-recall and medium-precision extractor, tweaked for β = 3 without GED features, that has a word-level precision, recall, and F_1 of about 58%, 87%, and 70% respectively for the mouse dataset and 48%, 78%, and 59% respectively for the fly dataset.

6 Related Work

The identification of gene-protein names has received substantial attention in the bioinformatics community. Some prior research involves training an extractor on weakly-labeled gene-protein synonyms; for instance, Hachey *et al.* [12] automatically labeled gene text fragments by identifying potential genes using regular expression fuzzy matching, and then trained a tagger for each organism. The most closely related prior work is that of Morgan *et al.* [22] perform pattern matching to

find candidate mentions in FlyBase abstracts using synonym lists and trained a HMM-based tagger on these noisy training data, achieving a F_1 of 67% with 522,825 tokens of training data and a F_1 of 75% with 1,342,039 tokens of training data.

There are several additional contributions of this work. Unlike Morgan *et al*, we study the generality of weak-labeling methods (our system is the same for FlyBase and MGI). We also study the use of intra-document repetition, and its effect on weakly-trained NER systems, alone and in combination with other methods. We also study the effect of sentence filtering, and the effect of GED (dictionary) features on the range of points reachable on a recall-precision curve. Our F_1 performance for the fly data with 1057 abstracts is comparable to that obtained by Morgan *et al*. with 522,825 tokens (approximately 2000-2500 abstracts). However, Morgan *et al* exploited orthographic preprocessing steps that we did not use, and the effect of using much larger training sets. (Unfortunately we cannot compare directly on the same test set, due to technical issues involving tokenization.)

Some other prior related research involves unsupervised identification of gene-protein names. Wellner [27] incorporates part-of-speech as factors for proposing gene phrases and performs exact matching from a synonym list to abstracts for annotating candidate gene-protein synonyms. Cohen [5] generates orthographic variants of gene-protein entities, separates out regular English words by using English word dictionaries, and matches the remaining variants against biomedical abstracts.

The contribution of this paper is to explore and systematically evaluate several different techniques, in isolation and in combination, for the gene-protein NER task: sentence filtering, GED features [24], SDR labeling [20], training on weakly-labeled examples [22], and tuning trained extractors [21]. We also contribute to the community, for each of fly and mouse organism, two organism-specific gene-protein name extractors[4]; one has high precision but medium recall and the other high recall but medium precision.

7 Conclusions

Manually annotated training data has always been difficult to produce. This is especially true for biomedical data, because expertise in biology is required to annotate gene-protein names. In this paper, we trained a gene-protein NER system, without manually annotating any documents, by utilizing the mouse and fly dataset from BioCreAtIvE task 1B. We presented an automatic approach for creating training corpora by soft matching gene synonyms into abstracts. We illustrated that the NER systems trained on these annotated abstracts, combined with sentence filtering, SDR labeling, and/or GED features, can outperform all baselines. Furthermore, we also demonstrated the possibility of converting a gene-protein NER system with decent performance into a high-recall gene-protein name extractor. Our results demonstrate that the quality of named entity recognition systems can be significantly improved through the use of readily available data and thus avoiding the difficult process of manually annotating training sets.

Acknowledgements

This material is based upon work supported by supported by NIH K25 grant DA017357-0, and the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NIH, the Defense Advanced Research Projects Agency (DARPA), or the Department of Interior-National Business Center (DOI-NBC).

Appendix A. Stop-Words

List of common English words that are used as stop-words in our system: *all*, *an*, *and*, *are*, *as*, *at*, *between*, *but*, *by*, *can*, *for*, *from*, *has*, *in*, *into*, *is*, *it*, *less*, *likely*, *more*, *most*, *much*, *not*, *of*, *on*, *or*, *per*, *such*, *that*, *the*, *through*, *to*, *via*, *was*, *we*, *were*, *whereas*, *whole*, *with*.

[4] Available from http://rcwang.com/pub/GeneNER.tar.gz

References

[1] **FlyBase: A database of the drosophila genome** [http://flybase.bio.indiana.edu]

[2] **MGI: mouse genome informatics** [http://www.informatics.jax.org]

[3] R. Bunescu, R. J. Mooney: **Relational markov networks for collective information extraction.** In *ICML-2004 Workshop on Statistical Relational Learning*; 2004.

[4] Razvan Bunescu, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun Kumar Ramani, Yuk Wah Wong: **Learning to extract proteins and their interactions from medline abstracts.** In *Proceedings of the ICML-2003 Workshop on Machine Learning in Bioinformatics*. pp. 46-53. Washington DC; 2003:46-53.

[5] Aaron M. Cohen: **Unsupervised gene/protein named entity normalization using automatically extracted dictionaries.** In *Proceedings of the ACL-ISMB Workhop on Linking Biological Literature, Ontologies and Databses: Mining Biological Semantics*. pp. 14-24. Detroit; 2005:14-24.

[6] William W. Cohen: **Minorthird: Methods for identifying names and ontological relations in text using heuristics for inducing regularities from data** [http://minorthird.sourceforge.net]

[7] N. Collier, H. Park, N. Ogata, Y. Tateisi, C. Nobata, T. Ohta, T. Sekimizu, H. Imai, K. Ibushi, J. Tsujii: **The GENIA project: Corpus-based knowledge acquisition and information extraction from genome research papers.** In *Proceedings of EACL-99*. pp. 271-272; 1999:271-272.

[8] Michael Collins: **Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms.** In *Empirical Methods in Natural Language Processing (EMNLP)*; 2002.

[9] Mark Craven, Johan Kumlien: **Constructing biological knowledge bases by extracting information from text sources.** In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB-99)*. pp. 77-86: AAAI Press; 1999:77-86.

[10] Kristofer Franzén, Gunnar Eriksson, Fredrik Olsson, Lars Asker, Per Lidén, Joakim Cöster: **Protein names and how to find them.** *International Journal of Medical Informatics* 2002, **67:**49-61.

[11] K. Fukuda, T. Tsunoda, A. Tamura, T. Takagi: **Toward information extraction: Identifying protein names from biological papers.** In *Proceedings of the Pacific Symposium on Biocomputing '98 (PSB-98)*. pp. 707-718; 1998:707-718.

[12] Ben Hachey, Huy Nguyen, Malvina Nissim, Bea Alex, Claire Grover: **Grounding Gene Mentions with Respect to Gene Database Identifiers.** In *BioCreAtIvE Workshop Handouts*. Granada, Spain; 2004.

[13] Daniel Hanisch, Katrin Fundel, Heinz-Theodor Mevissen, Ralf Zimmer, Juliane Fluck: **ProMiner: rule-based protein and gene entity recognition.** *BMC Bioinformatics* 2005, **6**.

[14] Lynette Hirschman, Marc Colosimo, Alexander Morgan, Alexander Yeh: **Overview of BioCreAtIvE task 1B: normalized gene lists.** *BMC Bioinformatics* 2005, **6**.

[15] Lynette Hirschman, Alexander Yeh, Christian Blaschke, Alfonso Valencia: **Overview of BioCreAtIvE: critical assessment of information extraction for biology.** *BMC Bioinformatics* 2005, **6**.

[16] K. Humphreys, G. Demetriou, R. Gaizauskas: **Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures.** In *Proceedings of the Pacific Symposium on Biocomputing (PSB-2000)*. pp. 502-513; 2000:502-513.

[17] K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, Y. Wilks: **Univ. of Sheffield: Description of the LASIE-II system as used for MUC-7.** In *Message Understanding Conference Proceedings (MUC-7)*. Fairfax, Virginia; 1998.

[18] Zhenzhen Kou, William W. Cohen, Robert F. Murphy: **High-recall protein entity recognition using a dictionary.** *Bioinformatics* 2005, **21:**i266-i273.

[19] Vladimir I. Levenshtein: **Binary codes capable of correcting deletions, insertions, and reversals.** *Soviet Physics Doklady* 1966, **10:**707-710.

[20] Einat Minkov, Richard C. Wang, William W. Cohen: **Extracting personal names from emails: Applying named entity recognition to informal text.** In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*. pp. 443-450. Vancouver, B.C., Canada; 2005:443-450.

[21] Einat Minkov, Richard C. Wang, Anthony Tomasic, William W. Cohen: **NER Systems that suit user's preferences: Adjusting the recall-precision trade-off for entity extraction.** In *Human Language Technology Conference - North American Chapter of the ACL (HLT-NAACL)*. New York City; 2006.

[22] Alexander A. Morgan, Lynette Hirschman, Marc Colosimo, Alexander S. Yeh, Jeff B. Colombe: **Gene name identification and normalization using a model organism database.** *Journal of Biomedical Informatics* 2004, **37:**396-410.

[23] Robert F. Murphy, Zhenzhen Kou, Juchang Hua, Matthew Joffe, William W. Cohen: **Extracting and Structuring Subcellular Location Information from On-line Journal Articles: The Subcellular Location Image Finder.** In *KSCE*; 2004.

[24] Sunita Sarawagi, William W. Cohen: **Semi-markov conditional random fields for information extraction.** In *NIPS*; 2004.

[25] Charles Sutton, Andrew McCallum: **Collective segmentation and labeling of distant entities in information extraction.** In *ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields*. Banff, Canada; 2004.

[26] Andreas Vlachos, Caroline Gasperin: **Bootstrapping and Evaluating Named Entity Recognition in the Biomedical Domain.** In *Proceedings of the HLT-NAACL Workshop on Linking Natural Language Processing and Biology*. Brooklyn, New York; 2006.

[27] Ben Wellner: **Weakly Supervised Learning Methods for Improving the Quality of Gene Name Normalization Data.** In *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*. pp. 1-8. Detroit; 2005:1-8.